Extraction of biomedical events using case-based reasoning

Mariana L. Neves
Biocomputing Unit
Centro Nacional de Biotecnología - CSIC
C/ Darwin 3, Campus de Cantoblanco,
28049, Madrid, Spain
mlara@cnb.csic.es

José M. Carazo
Biocomputing Unit
Centro Nacional de Biotecnología - CSIC
C/ Darwin 3, Campus de Cantoblanco,
28049, Madrid, Spain
carazo@cnb.csic.es

Alberto Pascual-Montano
Departamento de Arquitectura de Computadores
Universidad Complutense de Madrid, Facultad de
Ciencias Físicas
28040, Madrid, Spain
pascual@fis.ucm.es

Abstract

The BioNLP´09 Shared Task on Event Extraction presented an evaluation on the extraction of biological events related to genes/proteins from the literature. We propose a system that uses the case-based reasoning (CBR) machine learning approach for the extraction of the entities (events, sites and location). The mapping of the proteins in the texts to the previously extracted entities is carried out by some simple manually developed rules for each of the arguments under consideration (cause, theme, site or location). We have achieved an f-measure of 24.15 and 21.15 for Task 1 and 2, respectively.

1 Introduction

The increasing amount of biological data generated by the high throughput experiments has lead to a great demand of computational tools to process and interpret such amount of information. The protein-protein interactions, as well as molecular events related to one entity only, are key issues as they take part in many biological processes, and many efforts have been dedicate to this matter. For example, databases are available for the storage of such interaction pairs, such as the Molecular INTeraction Database (Chatr-aryamontri et al., 2007) and IntAct (Kerrien et al., 2007).

In the field of text mining solutions, many efforts have been made. For example, the BioCreative II protein-protein interaction (PPI) task (Krallinger, Leitner, Rodriguez-Penagos, & Valencia, 2008) consists of four sub-tasks, including the extraction of the protein interaction pairs in full-text documents, achieving an f-measure of up to 0.30. The initiative of annotation of both Genia corpus (J. D. Kim, Ohta, & Tsujii, 2008) and BioInfer (Pyysalo et al., 2007) is another good example.

The BioNLP´09 Shared Task on Event Extraction (J.-D. Kim, Ohta, Pyysalo, Kano, & Tsujii, 2009) proposes a comparative evaluation for the extraction of biological events related to one or more gene/protein and even other types of entities related to the localization of the referred event in the cell. The types of events that have been considered in the shared task were localization, binding, gene expression, transcription, protein catabolism, phosphorylation, regulation, positive regulation and negative regulation. A corpus that consisted of 800, 150 and 260 PubMed documents (title and abstract text only) was made available for the training, development test and testing datasets, respectively. For all documents, the proteins that took part in the events were provided.

The shared task organization proposed three tasks. Task 1 (Event detection and characterization) required the participants to extract the events from the text and map them to its respec-

Proceedings of the Workshop on BioNLP: Shared Task, pages 68–76,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

tive theme(s), as an event may be associated to one or more themes, e.g. binding. Also, some events may have only a gene/protein as theme, e.g. protein catabolism, while some other may be also associated to another event, e.g. regulation events. Task 2 (Event argument recognition) asked the participants to provide the many arguments that may be related to the extracted event, such as its cause, that may be an annotated or one of the previously extracted events. Other arguments include site and localization, which should be first extracted from the texts by the system, as they do not come annotated in the documents. Task 3 (Recognition of negation and speculations) evaluates the presence of negations and speculation related to the previously extracted events.

Our group has participated in this shared task with a system implemented with the case-based reasoning (CBR) machine learning technique as well as some manual rules. We have presented results for tasks 1 and 2 exclusively. The system described here is part of the Moara project[1] and was developed in Java programming language and use MySQL database.

2 Methods

Case-based reasoning (CBR) (Aamodt & Plaza, 1994) is the machine learning method that was used for extracting the terms and events here proposed and consists of first learning cases from the training documents, by means of saving them in a base of case, and further retrieving a case the most similar to a given problem during the testing step, from which will be given the final solution, hereafter called “case-solution”. One of the advantages of the CBR algorithm is the possibility of getting an explanation of why to a given token has been attributed a certain category, by means of checking the features that compose the case-solution. Additionally, and due to the complexity of the tasks, a rule-based post-processing step was built in order to map the previously extracted terms and events among themselves.

2.1 Retaining the cases

In this first step, documents of the training dataset are tokenized according to spaces and punctuations. The resulting tokens are represented in the CBR approach as cases composed of some predefined features that take into account the morphology and grammatical function of the tokens in the text as well as specific features related to the problem under consideration. The resulting cases are then stored in a base of case to be further retrieved (Figure 1).

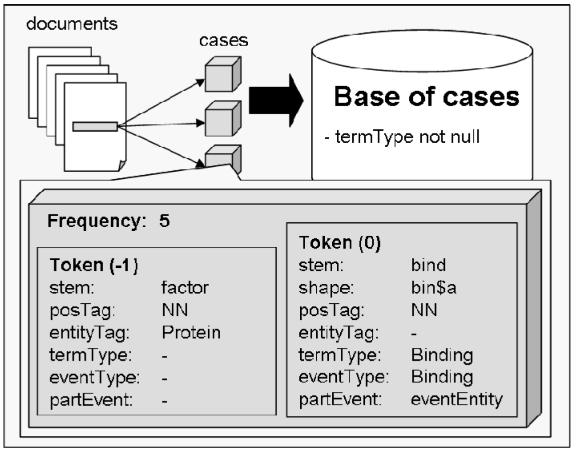


Figure 1: Training step in which cases are represented by some pre-defined features and further saved to a base.

Regarding the features that compose a case, these were the ones that were considered during the training and development phases: the token itself (token); the token in lower case (lowercase); the stem of the token (stem); the shape of the token (shape); the part-of-speech tag (posTag); the chunk tag (chunkTag); a biomedical entity tag (entityTag); the type of the term (termType); the type of the event (eventType); and the part of the term in the event (eventPart). The stem of a token was extracted using an available Java implementation[2] of the Porter algorithm (Porter, 1980), while the part-of-speech, chunk and bio-entity tags were taken from the GENIA Tagger (Tsuruoka et al., 2005).

The shape of a token is given by a set of characters that represent its morphology: “a” for lower case letters, “A” for upper case letters, “1” for numbers, “g” for Greek letters, “p” for stop-

[1] http://moara.dacya.ucm.es

[2] http://www.tartarus.org/~martin/PorterStemmer

words[3], “$” for identifying 3-letters prefixes or suffixes or any other symbol represented by itself. Here are some few example for the shape feature: “Dorsal” would be represented by “Aa”, “Bmp4” by “Aa1”, “the” by “p”, “cGKI(alpha)” by “aAAA(g)”, “patterning” by “pat$a” (‘$’ symbol separating the 3-letters prefix) and “activity” by “a$vity” (‘$’ symbol separating the 4-letters suffix). No repetition is allowed in the case of the “a” symbol for the lower case letters.

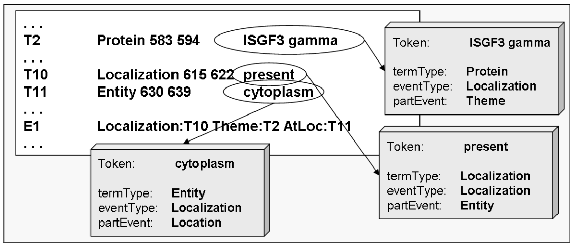


Figure 2: Example of the termType, eventType and partEvent features.

The last three features listed above are specific to the event detection task and were extracted from the annotation files (.a1 and .a2) that are part of the corpus. The termType feature is used to identify the type of the term in the event problem, and it is extracted from the term lines of both annotation files .a1 and .a2, i.e. the ones which the identifiers starts with a “T”. The eventType features represent the event itself and it is extracted from the event lines of .a2 annotation file, i.e. the ones that starts with an “E”. Finally, eventPart represents the token according to its role, i.e. entity, theme, cause, site and location. The termType, eventType and eventPart features are the hereafter called “feature-problem”, the features that are unknown to the system in the testing phase and which values are to be given by the case-solution. Figure 2 illustrate one example of these features for an extract of the annotation of the document “1315834” from the training dataset.

Usually, one case corresponds for each token of the documents in the training dataset. However, more than one case may be created from a token, as well as none at all, depending on the predefined features. For example, some tokens may derive in more than one case due to the shape feature, as for example, “patterning” (“pat$a”, “a$ing”, “a”). Also, according to the retaining strategy, some tokens may be associated to no case at all, for example, by restricting the value of a determined feature as the retaining strategy. In order to reduce the number of retained cases, and consequently reduce the further retrieving time, only those tokens related to an event are retained, i.e., tokens with not null value for the termType feature.

The text of a document may be read in the forward or backward direction during the training step, and even combining both of them (Neves, Chagoyen, Carazo, & Pascual-Montano, 2008). Here, we have considered the forward direction exclusively. Also, another important point is the window of tokens under consideration when setting the features of a case, if taking into account only the token itself or also the surrounding tokens, the ones which come before or after it. Here we consider a window of (-1,0), i.e., for each token, we get the feature of the token itself and of the preceding one, exclusively.

Features / Tokens	Training		Testing	
	-1	0	-1	0
stem	✓	✓	✓	✓
shape		✓		✓
posTag	✓	✓	✓	✓
chunkTag				
entityTag	✓	✓	✓	✓
termType	✓	✓	✓	✓
eventType	✓	✓	✓	
partEvent	✓	✓	✓	

Table 1: Selected features in the training and testing steps for the tokens “0” and “-1”. The last three features are the ones to be inferred.

Many experiments have been carried out in order to choose the best set of features (Table 1). The higher the number of features under consideration, the greater is the number of cases to be retained and the higher is the time needed to search for the case-solution. He relies therefore the importance of choosing a small an efficient set of features. For this reason, the shape features has not been considered for the preceding token (-1) in order to reduce the number of cases, as this shape usually result in more than one case per token. The termType feature is at the same time known and unknown in the testing step. It is know for the protein terms but is un-

[3] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/

known for the remaining entities (events, sites and locations).

By considering these features for the 800 documents in the training set, about 26,788 unique cases were generated. It should be noted that no repetition of cases with the same values for the features are allowed, instead a field for the frequency of the case is incremented to keep track of the number of times that it has appeared during the training phase. The frequency range goes from 1 (more than 22,000 cases) to 238 (one case only).

2.2 Retrieving a case

When a new document is presented to the system, it is first read in the forward direction and tokenized according to space and punctuation and the resulting tokens are mapped to cases of features, exactly as discussed in the retaining step. The only difference here is the set of feature (cf. Table 1), as some of them are unknown to the system and are the ones to be inferred from the cases retained during the training step.

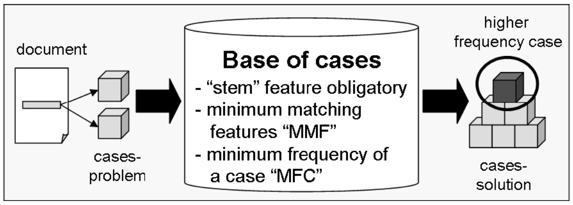


Figure 3: Retrieval procedure to choose the most case-solution with higher frequency and based on MMF and MFC parameters.

For each token, the system first creates a case (hereafter called "case-problem") based on the testing features and proceeds to search the base of cases for the case-solution the most similar to this case-problem (Figure 3). It should be noted that a token may have more than one case-problem, depending of the values of the shape feature. The best case-solution among the ones found by the system will be the one with the higher frequency. The system always tries to find a case-solution with the higher number of features that have exactly the same value of the case-problem's respective features. The stem is the only mandatory feature which value must be always matched between the case-problem and the case-solution. The value of the two features-problem (eventType and partEvent) will be given by the values of the case-solution's respective features. If no case solution is found, the token is considered of not being related to the event domain in none of its parts (entity, theme, cause, etc.).

Two parameters have been taken into consideration in the retaining strategy: the minimum matching feature (MMF) and the minimum frequency of the case (MFC). The first one set the minimum features that should be matched between the case-problem and the case-solution, as the higher the number of equal features between theses cases, the more precise is the decision inferred from the case-solution.

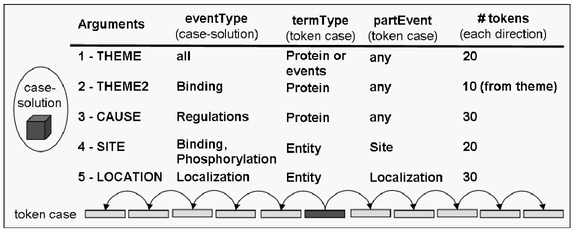
On the other hand, the MFC parameter restricts the cases that are to be considered by the search strategy, the ones with frequency higher than the value specified by this parameter. The higher the minimum frequency asked for a case, the lower is the number of cases under consideration and the lower is the time for obtaining the case-solution. From the 26,788 cases we have retained during the training phase, about 22,389 of them appeared just once and would not be considered by the searching procedure if the MFC parameter was set to 2, for example, therefore reducing the searching time.

Experiments have been carried out in order to decide the values for both parameters and it resulted that a better performance is achieved (cf. 3) by setting the MFC to a value higher than 1. On the other hand, experiments have shown that the recall may decrease considerably when restricting the MMF parameter.

By repeating this procedure for all the tokens of the document, the latter may be then considered as being tagged with the event entities. However, in order to construct the output file required by the shared task organization, some manual rules have been created in order to map the events mapped to its respective arguments, as described in the next section.

2.3 Post-processing rules

For the tasks 1 and 2, the participants were asked to output the events present in the provided texts along with their respective arguments. The events have been already extracted in the previous step; the tokens that were tagged as "Entity" for the "partEvent" feature (cf. Figure 2), hereafter called "event-entity". This entity is the start point from which to search for the arguments which are incrementally extracted from the text in the following order: theme, theme 2, cause, site and location. Figure 4 resumes the rules for each of the arguments.



Arguments	eventType (case-solution)	termType (token case)	partEvent (token case)	# tokens (each direction)
1 - THEME	all	Protein or events	any	20
2 - THEME2	Binding	Protein	any	10 (from theme)
3 - CAUSE	Regulations	Protein	any	30
4 - SITE	Binding, Phosphorylation	Entity	Site	20
5 - LOCATION	Localization	Entity	Localization	30

Figure 4: Resume of the post-processing rules for each type of argument.

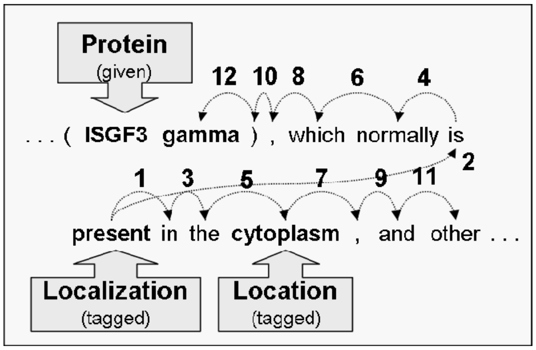
Themes: The theme-candidates for an event-entity are the annotated proteins (.a1 file) as well as the events themselves, in the case of the regulation, positive regulation and negative regulation events. The first step is then to try to map each event to its theme and in case that no theme is found, the event is not considered anymore by the system and it is not printed to the output file.

The theme searching strategy starts from the event-entity and consists of reading the text in both directions alternatively, one token in the forward direction followed by one token in the backward direction until a theme-candidate is found (Figure 5). The system halts if the end of the sentence is found or if the specified number of tokens in each direction is reached, 20 for the theme. By analyzing some of the false negatives returned from the experiments with the development dataset, we have learned that few events are associated to themes present in a different sentence and although aware these cases, we have decided to restrict the searching to the sentence boundaries in order to avoid a high number of false positives.

In the case of a second theme, allowed for binding events only, a similar searching strategy is carried out, except that here the system reads up of 10 tokens in each direction, starting from the theme entity previously extracted.

Cause: The cause-candidates are also the annotated proteins and, starting from the event-entity, a similar search is carried out, restricted up to 30 tokens in each direction and to the boundaries of the same sentence. This procedure is carried out for the regulation, positive regulation and negative regulation events only and the only extra restriction is that the candidate should not be the protein already assigned as theme. If no candidate is found, the system considers that there is no cause associated to the event under consideration.

Site and Location: Here the candidates are the tokens tagged with the values of "Entity" for the termType feature, and "Site" and "Location" for the partEvent feature, respectively. The search for the site is carried out for the binding and phosphorylation events and the location search for the localization event only. The procedure is restricted to the sentence boundaries and up to 20 and 30 tokens, respectively, starting from the event-entity. Once again, if not candidate is found, the system consider that there is no site or location associated to the event under consideration.



Protein
(given)
12 10 8 6 4
. . . (ISGF3 gamma) , which normally is
2
1 3 5 7 9 11
present in the cytoplasm , and other . . .
Localization
(tagged)
Location
(tagged)

Figure 5: Contribution of each class of error to the 275 false positives analyzed here.

3 Results

This section presents the results of the experiments carried out with the development and the blind test datasets as well as an analysis of the false negatives and false positives. Results here will be presented for tasks 1 and 2 in terms of precision, recall and f-measure.

Experiments have been carried out with the development dataset in order to decide the best value of the MMF and MFC parameters (cf. 2.2). Figure 6 shows the variation of the F-measure according to both parameters for the values of 1, 3, 4, 5, 6, 7 and 8 for MMF; and 1, 2, 5, 10, 15, 20 and 50 for MFC.

Usually, recall is higher for a low value of MFC, as the searching for the case-solution is carried out over a greater number of cases and the possibility of finding a good case-problem is higher. On the other hand, precision increases when few cases are under considered by the search strategy, as fewer decisions are taken and the cases-solution have usually a high frequency, avoiding decision based on "weak" cases of frequency 1, for example.

Figure 6 shows that the best value for MFC ranges from 2 to 20 and for MMF from 5 to 7 and the best f-measure result is found for the values of 2 and 6 for these parameters (f2m6), respectively. As these experiments have been carried out after the deadline of the test dataset, the run that was submitted as the final solution was the one with the values of 2 and 1 for the MFC and MMF parameters (f2m1), respectively. Table 3 and 4 resumes the results obtained for the test dataset with the configuration that was submitted (f2m1), and the best one (f2m6) after accomplishing the experiments above described. Results have slightly improved by only trying to choose the best values for the parameters here considered.

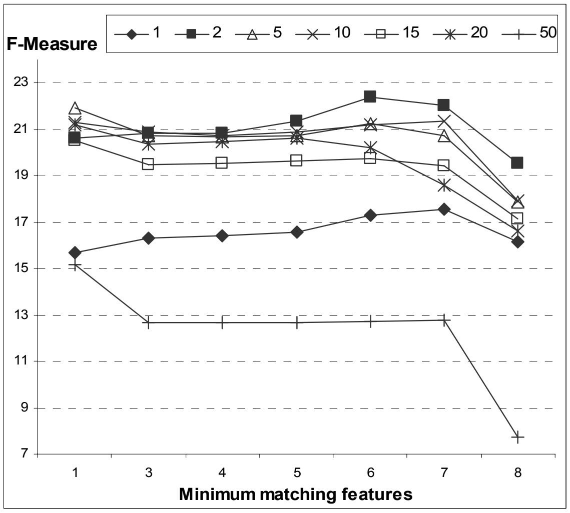


Figure 6: F-Measure for the development dataset in terms of the MFC (curves) and the MMF (x-axis).

An automatic analysis of the false positives and false negatives has been performed for the development dataset and for the results obtained with the final submission (f2m1), a total of 2502 false positives and 1300 false negatives. We have found out that the mistakes are related mainly to the retrieving of the case-solution and to the mapping of an event to its arguments. The mistakes have been classified in seven groups described below and figures 7 and 8 show the percent contribution of each class for the false positives and false negatives, respectively.

Events composed of more than one token (1): this mistake happens when the system is able to find the event with its correct type and arguments but with only part of its tokens, such as "regulation" instead of "up-regulation" and "reduced" or "levels" instead of "reduced levels", both in document 10411003. This is mainly due to our tokenization strategy of separating the tokens according to all punctuation and symbols (including hyphens) and also due to the evaluation method that seems not consider alternatives to the text of an event. This mistake always results in one false positive and one false negative.

tasks / results		**recall**	**precision**	**f-measure**
task 1	**(f2m1)**	28.63	20.88	24.15
	(f2m6)	27.18	23.92	25.45
task 2	**(f2m1)**	25.02	18.32	21.15
	(f2m6)	24.49	21.63	22.97

Table 3: Results for the test dataset (tasks 1 and 2).

Results / Events	**(f2m1)**			**(f2m6)**		
	p	**r**	**fm**	**p**	**r**	**fm**
prot. catab.	78.6	55.0	64.7	71.4	55.6	65.5
phosphoryl.	49.6	56.1	52.7	46.0	55.2	50.2
transcript.	48.9	19.8	28.1	38.7	29.6	33.5
neg. reg.	9.8	7.9	8.8	7.9	7.7	7.8
pos. reg.	10.0	6.6	7.9	10.2	8.0	9.0
regulation	8.6	4.5	5.9	7.5	5.3	6.3
localizat.	28.2	42.9	34.0	23.3	48.9	33.3
gene expr.	51.8	55.1	53.4	52.6	61.2	56.6
binding	19.5	12.1	14.9	22.4	14.4	17.5

Table 4: Results by event for Task 2 on test dataset.

Events and arguments in different sentences of the text (2): as we already discussed in section 2.3, our arguments searching strategy is restricted to the boundaries of the sentence. Some examples of this mistake may be found in document 10395645 in which two events of the token "activation [1354-1364]" is mapped to the themes "caspase-6 [1190-1199]" and "CPP32 [1165-1170]", both located in a different sentence. This mistake usually affects only the false negatives but may cause also a false positive if the system happens to find a valid (wrong) argument in the same sentences for the event under consideration.

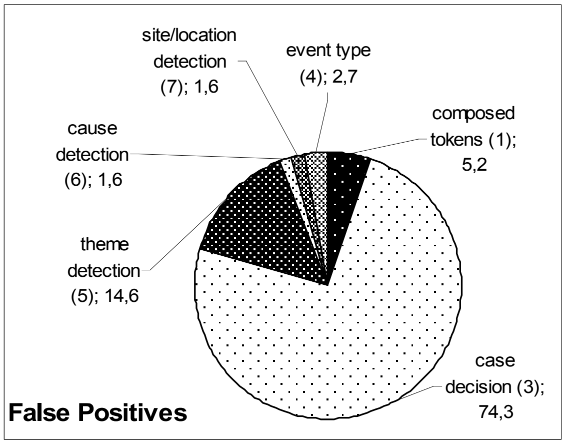


Figure 7: Percent contribution of each error to the false positives.

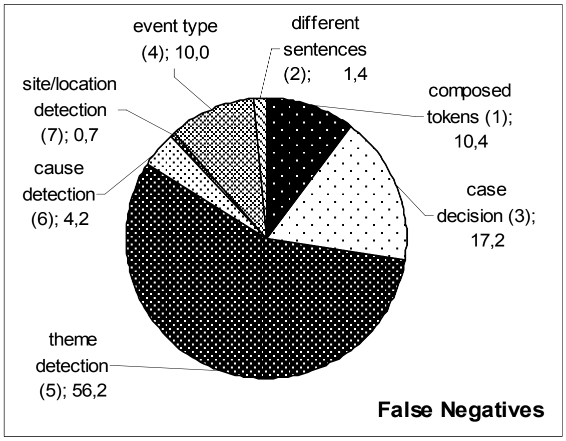


Figure 8: Percent contribution of each error to the false negatives.

Decision for a case (3): this class of error is due to the selection of a wrong case-solution and we include in this class mistakes due to two situations: when the system fails to find any case-solution for an event token (false negative) or when a case-solution is found for a non-event token (false positive). The first situation is only dependent of the searching strategy and its two parameters (MMF and MFC) while the second one is also related to the post-processing step, if the latter succeeds to find a theme for the incorrectly extracted event. An example of a false negative that falls in this group is “dysregulation [727-740]” from document 10229231 that failed to be mapped to a case-solution. Regarding the false positives, this class of mistake is the majority of them and it is due to the low precision of the system that frequently is able to find cases-solution associated to tokens that are not events at all, such as the token “transcript [392-402]” of document 10229231. It should be noted that the incorrect association of a token to a case-solution does not result in a false positive a priori, but only if the post-processing step happen to find a valid theme to it, a mistake further described in group 5.

Wrong type of the event (4): this class of mistake is also due to the wrong selection of a case-solution, but the difference here is that the token is really an event, but the case-solution is of the wrong type, i.e. it has a wrong value for the eventType feature. The causes of this mistake are many, such as, the selection of features (cf. Table 1) or the value of the MFC parameter that may lead to the selection of a wrong but more frequent case. We also include in this group the few false negatives mistakes in which a token is associated to more than one type of event in the gold-standard, such as the token “Overexpression [475-489]” from document 10229231 that is associated both to a Gene Expression and to a Positive Regulation event. One way of overcome it would be to allow the system to associated more than one case to a token, taking the risk of decreasing the precision.

Theme detection (5): in this group falls more than half of the false negatives and we include here only those mistakes in which the token was correctly associated to a case-solution of the correct type. These mistakes may be due to a variety of situations related to the theme detection, such as: the association of the event to another event when it should have been done to a protein or vice-versa (for the regulation events); the mapping of a binding event to one theme only when it should have been two theme or vice-versa; the association of the event to the wrong protein theme, especially when there is more than one nearby; and even not being able to find any theme at all. Also, half of theses mistakes happen when an event is associated to more than one theme separately, not as a second theme. For example, the token “associated [278-288]”, from document 10196286, is associated in the gold standard to three themes – “tumor necrosis factor receptor-associated factor (TRAF) 1 [294-351]”, “2 [353-354]” and “3 [359-360]” – and we were only able to extract the first of them. This is due to the fact that we restrict the system to search only one “first” and one “second” theme for each event.

Cause detection (6): similar to the previous class, these mistakes happens when associating a cause to an event (regulation events only) when there is no cause related to it or vice-versa. For example, in document 10092805, the system has correctly mapped the token “decreases [1230-1239]” to the theme “4E-BP1 [1240-1246]” but also associated to it an inexistent cause “4E-BP2 [1315-1321]”. The evaluation of Task 2 does not allow the partial evaluation of an event and therefore a false positive and a false negative would be returned for the example above.

Site/Location detection (7): this error is similar to the previous one but related only to binding, phosphorylation and localization events, when the system fails to associate a site or a location to an event or vice-versa. For example, in document 10395671, the token “phosphorylation [1091-1106]” was correctly mapped to the theme “Janus kinase 3 [1076-1090]” but was also associated to an inexistent site “DNA [1200-1203]”. Once again, the evaluation of Task 2 does not allow the partial evaluation of the event and a false positive and a false negative would be returned.

We have also carried out an evaluation of our own in order to check the performance of our system only on the extraction the entities (event, site and location), not taking into account the association to the arguments. Table 5 resumes the values of precision, recall and f-measure for each type of term. The high recall confirm that most of the entities were successful extracted although the precision is not always satisfactory, proving that the tagging of the entities is not as hard a task as it is the mapping of the arguments. Additional results and more a detailed analysis of the errors may be found at Moara page[4].

4 Conclusions

Results show that our system has performed relatively well using a simple methodology of a machine learning based extraction of the entities and manual rules developed for the post-processing step. The analysis of the mistakes presented here confirms the complexity of the tasks proposed but not the extraction of the event terms (cf. Table 5).

We consider that the part of our system that requires most our attention is the retrieval of the case-solution and the theme detection of the post-processing step, in order to increase the precision and recall, respectively. The decision of searching for a second theme and of associating a single event separately to more than one theme is hard to be accomplished by manual rules and could better be learned automatically using a machine learning algorithm.

Events	**(f2m1)**			**(f2m6)**		
	p	**r**	**fm**	**p**	**r**	**fm**
prot. catab.	70.8	89.5	79.1	69.6	84.2	76.2
phosphoryl.	75.0	94.7	83.7	79.1	89.5	84.0
transcript.	22.7	75.9	34.9	36.4	74.6	48.9
neg. reg.	26.4	56.5	36.0	25.3	43.5	32.0
pos. reg.	24.3	63.7	35.2	26.5	59.1	36.6
regulation	20.8	65.9	31.7	22.1	52.5	31.1
localizat.	47.7	79.5	59.6	49.1	66.7	56.5
gene expr.	46.5	83.4	59.7	50.8	80.2	62.2
binding	29.7	71.1	41.9	29.7	64.4	40.7
entity	12.5	55.3	20.4	16.8	50.0	25.1
TOTAL	**27.5**	**69.2**	**39.4**	**30.9**	**62.9**	**41.4**

Table 5: Evaluation of the extraction of the event and site/location entities for the development dataset.

The automatic analysis of the false positive and false negative mistakes is a hard task since no hint is given for the reason of the mistake by the evaluation system, if due to the event type or to wrong theme, an incorrectly association to an event or even a missing cause or site.

Acknowledgments

This work has been partially funded by the Spanish grants BIO2007-67150-C03-02, S-Gen-0166/2006, PS-010000-2008-1, TIN2005-5619. APM acknowledges the support of the Spanish Ramón y Cajal program. The authors acknowledge support from Integromics, S.L.

References

Aamodt, A., & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications, 7*(1), 39-59.

[4] http://moara.dacya.ucm.es/results_shared_task.html

Chatr-aryamontri, A., Ceol, A., Palazzi, L. M., Nardelli, G., Schneider, M. V., Castagnoli, L., et al. (2007). MINT: the Molecular INTeraction database. *Nucleic Acids Res, 35*(Database issue), D572-574.

Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., et al. (2007). IntAct--open source resource for molecular interaction data. *Nucleic Acids Res, 35*(Database issue), D561-565.

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., & Tsujii, J. i. (2009). *Overview of BioNLP'09 Shared Task on Event Extraction.* Paper presented at the Proceedings of Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop, Boulder, CO, USA.

Kim, J. D., Ohta, T., & Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics, 9*, 10.

Krallinger, M., Leitner, F., Rodriguez-Penagos, C., & Valencia, A. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biol, 9 Suppl 2*, S4.

Neves, M., Chagoyen, M., Carazo, J. M., & Pascual-Montano, A. (2008). *CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem.* Paper presented at the Proceedings of the BioNLP 2008 Workshop at ACL 2008, Columbus, OH, USA.

Porter, M. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130-137.

Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J., et al. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics, 8*, 50.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., et al. (2005). *Developing a Robust Part-of-Speech Tagger for Biomedical Text.* Paper presented at the Advances in Informatics - 10th Panhellenic Conference on Informatics.